Introduction to Cryptology

11.3 - Factoring-based Digital Signatures

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

RSA Signatures: Textbook RSA

The plain RSA signature $S = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$ is defined as follows:

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(1)$: it runs a GenRSA algorithm on input a security parameter n. Then PK is set to (N, e), while SK is set to (N, d).[1].
- $\sigma \leftarrow \mathrm{Sign}(\mathrm{SK}, m)$: it takes a secret key (N, d), a message $m \in \mathbb{Z}_N^*$ and returns the signature $\sigma := m^d$.
- $1/0 \leftarrow \mathrm{Verify}(\mathrm{PK}, m, \sigma)$: on input a public key (N, e), a message m and a signature σ, it returns 1 if m is equal to σ^e, 0 otherwise.

---

[1] We recall that $N = pq$, where p and q are two distinct n-bit odd primes, while $[e]_{\varphi(N)}[d]_{\varphi(N)} = [1]_{\varphi(N)}$

Security of Textbook RSA

The RSA assumption relative to GenRSA implies hardness of forging signatures for a uniform message m.

Security of Textbook RSA

The RSA assumption relative to GenRSA implies hardness of forging signatures for a uniform message m.

- What about forgeries for messages chosen by $\mathcal{A}$?
- What if $\mathcal{A}$ can learn signatures on other messages?

Security of Textbook RSA

The RSA assumption relative to GenRSA implies hardness of forging signatures for a uniform message m.

- What about forgeries for messages chosen by $\mathcal{A}$?
- What if $\mathcal{A}$ can learn signatures on other messages?

No message attack: given a public key (N, e), pick $\sigma \in \mathbb{Z}_N^*$, compute the message as $m := \sigma^e$ and output the forgery (m, σ).

Malleability: given two valid signatures σ_1, σ_2, for messages m_1 and m_2, $\sigma_1 \cdot \sigma_2$ is a valid signature for $m = m_1 \cdot m_2$.

RSA-Full Domain Hash (RSA-FDH)

The RSA-Full Domain Hash signature

$$\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$$

is defined as follows.

- $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathrm{KeyGen}(n)$: it runs a GenRSA algorithm on input n and identifies a function $H : \{0,1\}^* \to \mathbb{Z}_N^*$. It then sets PK to (N, e, H) and SK to (N, d, H).
- $\sigma \leftarrow \mathrm{Sign}(\mathrm{SK}, m)$: on input a secret key (N, d, H) and a message $m \in \{0,1\}^*$, it returns $\sigma := H(m)^d$.
- $1/0 \leftarrow \mathrm{Verify}(\mathrm{PK}, m, \sigma)$: it takes a public key (N, e, H), a message m and a signature σ, and returns 1 if $H(m) = \sigma^e$, 0 otherwise.

Security of RSA-FDH

Theorem

If the RSA problem is hard relative to GenRSA *and* H *is modelled as a random oracle, then the digital signature RSA-FDH is existentially unforgeable.*

Security of (RSA-FDH)

Proof.

Let $\mathcal{A}$ be a PPT adversary against the $\mathrm{Sig}^{\mathrm{forge}}_{\mathcal{A},\Pi}(n)$ experiment.

We make the following assumptions:

- if $\mathcal{A}$ queries the signing oracle on a message m, then they previously queried H on m;
- the same is assumed for m^* in the forgery (m^*, σ^*);
- $\mathcal{A}$ makes exactly $q(n)$ distinct queries to H.

$\mathcal{A}$ is exploited as a subroutine to construct an adversary $\mathcal{A}'$ against the $\mathrm{RSA} - \mathrm{inv}_{\mathcal{A},\mathrm{GenRSA}}(n)$ experiment.

Security of RSA-FDH

$\mathcal{A}'$ receives (N, e, y) and manages a table.

- They choose a uniform element j in $\{1, \cdots, q\}$.
- They send PK $= (N, e)$ to $\mathcal{A}$.
- Hash queries: when $\mathcal{A}$ makes its i-th query m_i, $\mathcal{A}'$ replies as follows:
    - if $i = j$, the answer is y;
    - otherwise, a uniform σ_i is sampled in $\mathbb{Z}_N^*$, $y_i := \sigma_i^e$ is computed and (m_i, σ_i, y_i) is stored in the table. Then y_i is returned.

Security of RSA-FDH

- Signing queries: when $\mathcal{A}$ makes a signing query on m, by hypothesis $m = m_i$ for some m_i already in the table. Then $\mathcal{A}'$ replies as follows:
  - if $i = j$, they abort;
  - otherwise they find the entry (m_i, σ_i, y_i) in the table, and return σ_i to $\mathcal{A}$.
- If $\mathcal{A}$'s forgery (m^*, σ^*) is valid and m^* is equal to m_j, then $\mathcal{A}'$ outputs σ^*.

To conclude, we observe that

$$\Pr(\mathrm{RSA} - \mathrm{inv}_{\mathcal{A}',\mathrm{GenRSA}}(n) = 1) = \frac{\Pr(\mathrm{Sig}^{\mathrm{forge}}_{\mathcal{A},\Pi}(n) = 1)}{q(n)} \leq \mathrm{negl}(n)$$

$\square$

RSA-FDH

A signature scheme that can be viewed as a variant of RSA-FDH is included in the RSA PKCS #1 v2.1 standard.

Practical attacks on RSA-FDH are known if H has a small output length (the range of H should be close to all $\mathbb{Z}_N^*$).

Hash functions such as SHA-1 are not suitable.

Further Reading I

- Carlisle Adams and Steve Lloyd.
  Understanding PKI: concepts, standards, and deployment considerations.
  Addison-Wesley Professional, 2003.

- Dan Boneh, Ben Lynn, and Hovav Shacham.
  Short signatures from the Weil pairing.
  Journal of cryptology, 17(4):297–319, 2004.

- Tim Dierks.
  The transport layer security (TLS) protocol version 1.2.
  2008.

- Carl Ellison and Bruce Schneier.
  Ten risks of PKI: What you're not being told about public key infrastructure.
  Comput Secur J, 16(1):1–7, 2000.

Further Reading II

- Amos Fiat and Adi Shamir.
  How to prove yourself: Practical solutions to identification and signature problems.
  In Advances in Cryptology—CRYPTO'86, pages 186–194. Springer, 1987.

- Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov.
  The most dangerous code in the world: validating SSL certificates in non-browser software.
  In Proceedings of the 2012 ACM conference on Computer and communications security, pages 38–49. ACM, 2012.

Further Reading III

Hugo Krawczyk.
Cryptographic extraction and key derivation: The HKDF scheme.
In Annual Cryptology Conference, pages 631–648. Springer, 2010.

Hugo Krawczyk, Kenneth G Paterson, and Hoeteck Wee.
On the security of the TLS protocol: A systematic analysis.
In Advances in Cryptology–CRYPTO 2013, pages 429–448. Springer, 2013.

Leslie Lamport.
Constructing digital signatures from a one-way function.
Technical report, Technical Report CSL-98, SRI International Palo Alto, 1979.